Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

Информатики и радиоэлектроники»

Учебная дисциплина: «Программирование мобильных информационных систем»

Отчет

по выполнению лабораторной работы «Основы языка Kotlin»

Выполнил студент: гр. 210101 Жарский Е.А.

Проверил: Усенко Ф.В.

Минск 2024

**1. Цель работы:** изучить основные конструкции языка Kotlin, научиться работать с типами данных, операциями ввода-вывода, а также условными операторами. Закрепить полученные знания через выполнение задач, требующих практического применения теории.

**2. Задание:** продвинутый калькулятор ИМТ с историей и графиками: Создайте программу, которая вычисляет ИМТ, сохраняет историю всех вычислений и строит график изменения ИМТ с течением времени. Программа должна поддерживать экспорт данных в текстовый файл.

**3. Оснащение работы:** техническое задание, технический проект, ЭВМ, Kotlin, Itelij IDEA.

Файл Main.kt

import java.util.\*

fun main() {

val bmiHistory = mutableListOf<Pair<Date, Float>>()

while (true) {

println("Введите вес в килограммах (или введите 'exit' для завершения):")

val weightInput = readlnOrNull()

if (weightInput == "exit") break

val weight = weightInput?.toFloatOrNull()

println("Введите рост в метрах:")

val heightInput = readlnOrNull()

val height = heightInput?.toFloatOrNull()

if (weight != null && height != null) {

val bmi = BMICalculator.calculateBMI(weight, height)

val currentDate = Date()

println("Ваш ИМТ: %.2f".format(bmi))

println("Ваша текущая категория ИМТ: ${BMICalculator.getBMICategory(bmi)}")

BMICalculator.saveToHistoryFile(currentDate, bmi)

} else {

println("Неверный ввод. Попробуйте снова.")

}

WeightChart.plotBMIHistory()

}

HistoryExporter.exportHistoryToFile(bmiHistory)

}

Файл HistoryExporter.kt

import java.io.File

import java.text.SimpleDateFormat

import java.util.\*

object HistoryExporter {

fun exportHistoryToFile(bmiHistory: List<Pair<Date, Float>>) {

val file = File("bmi\_export.txt")

file.writeText("История ИМТ:\n")

bmiHistory.forEach { (date, bmi) ->

file.appendText("${SimpleDateFormat("dd-MM-yyyy HH:mm").format(date)} - ИМТ: %.2f\n".format(bmi))

}

println("История ИМТ успешно экспортирована в файл bmi\_export.txt")

}

fun loadHistoryFromFile(bmiHistory: MutableList<Pair<Date, Float>>) {

val file = File("bmi\_history.txt")

if (!file.exists()) {

println("Файл истории не найден -> Пустая история")

return

}

file.forEachLine { line ->

if (line.startsWith("История ИМТ:")) return@forEachLine

val parts = line.split(" - ИМТ: ")

if (parts.size == 2) {

val date = SimpleDateFormat("dd-MM-yyyy HH:mm").parse(parts[0])

val bmi = parts[1].replace(",", ".").toFloatOrNull()

if (bmi != null) {

bmiHistory.add(date to bmi)

}

}

}

println("История ИМТ успешно загружена из файла")

}

}

Файл BMICalculator.kt

import java.io.File

import java.text.SimpleDateFormat

import java.util.\*

object BMICalculator {

fun calculateBMI(weight: Float, height: Float): Float {

return weight / (height \* height)

}

fun saveToHistoryFile(date: Date, bmi: Float) {

val file = File("bmi\_history.txt")

file.appendText("${SimpleDateFormat("dd-MM-yyyy HH:mm").format(date)} - ИМТ: %.2f\n".format(bmi))

}

fun getBMICategory(bmi: Float): String {

return when {

bmi < 18.5 -> "Недостаточная масса тела"

bmi in 18.5..24.9 -> "Норма"

bmi in 25.0..29.9 -> "Избыточная масса тела"

else -> "Ожирение"

}

}

}

Файл WeightChart.kt

import HistoryExporter.loadHistoryFromFile

import org.knowm.xchart.SwingWrapper

import org.knowm.xchart.XYChart

import org.knowm.xchart.XYChartBuilder

import java.util.\*

object WeightChart {

fun plotBMIHistory() {

val bmiHistory = mutableListOf<Pair<Date, Float>>()

loadHistoryFromFile(bmiHistory)

println(bmiHistory)

val chart: XYChart = XYChartBuilder()

.width(800)

.height(600)

.title("Изменение ИМТ с течением времени")

.xAxisTitle("Дата")

.yAxisTitle("ИМТ")

.build()

chart.styler.setDatePattern("dd-MM-yyyy HH:mm")

chart.styler.isLegendVisible = true

val xData = bmiHistory.map { it.first.time.toDouble() }.toDoubleArray()

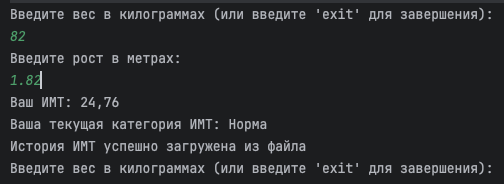
val yData = bmiHistory.map { it.second.toDouble() }.toDoubleArray()

chart.addSeries("ИМТ", xData, yData)

SwingWrapper(chart).displayChart()

}

}



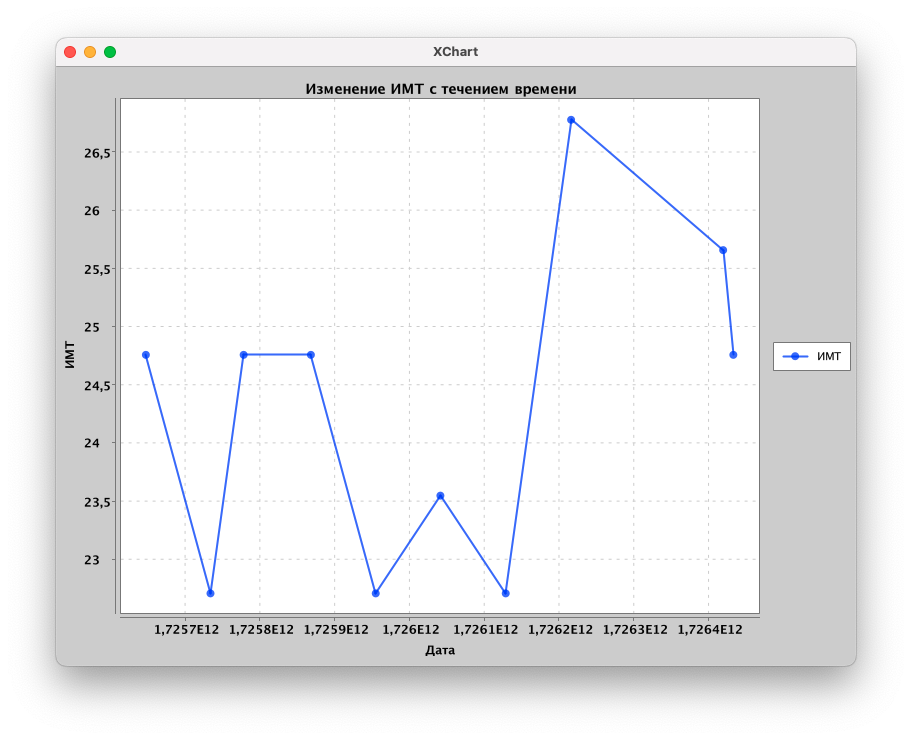
Рисунок 1 – Результат исполнения кода программы

Рисунок 2 – Построение графика изменения ИМТ